

# Lofar pipeline processing with Uniconore

## Work status

8. June 2016 | Stefan Fröhlich

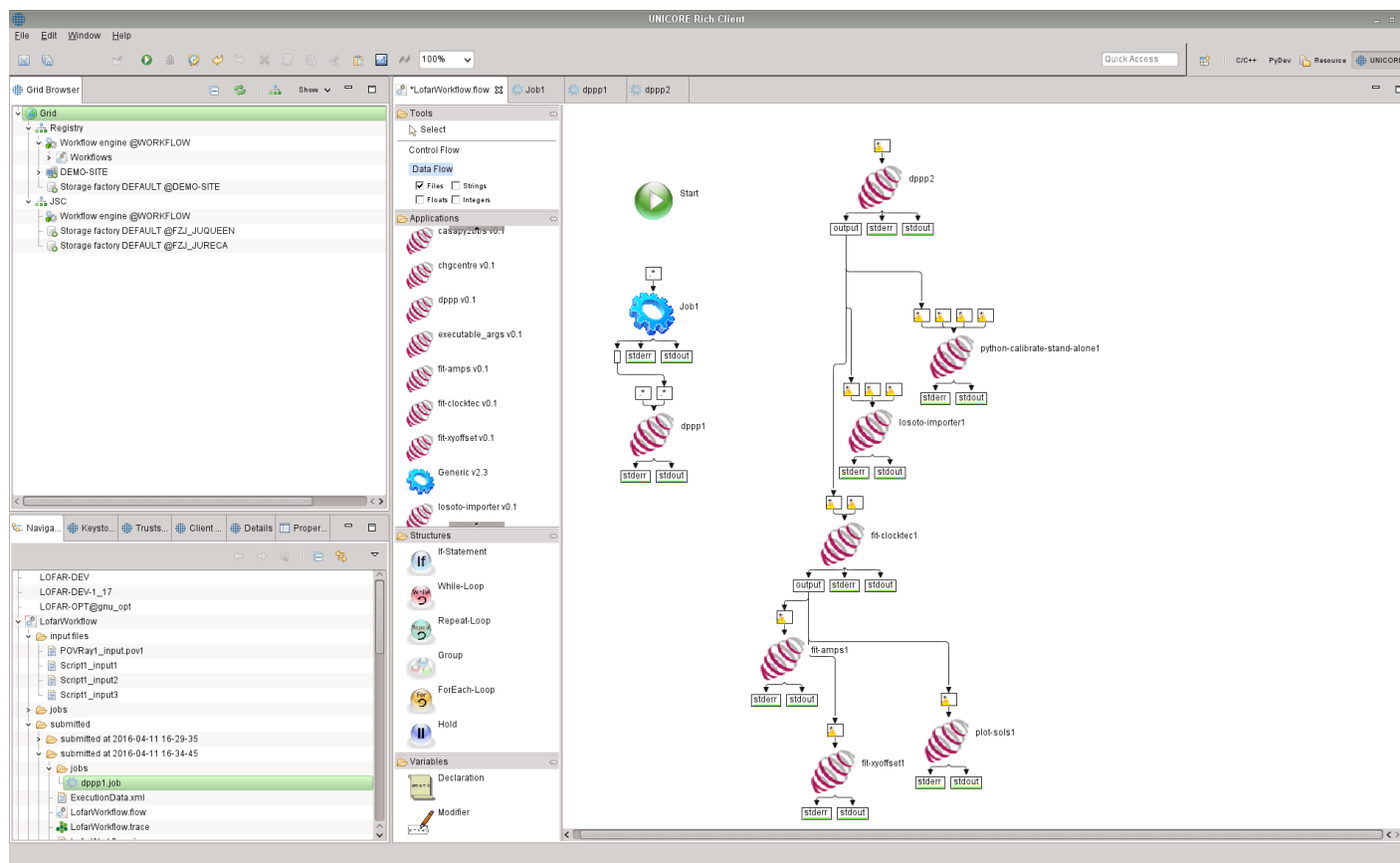
# Unicore

- Open source Grid Middleware
- Consists of Server and Client software
- Server
  - Grid services; job execution, storage access, file transfer
  - Interfaces for batch systems (LoadLeveler, Torque, SLURM)
- Client:
  - Eclipse based graphical client
  - Command line client
  - Web portal

## Unicore Rich Client

- Grid browser:  
Shows all available machines registered for your service.
- Workflow editor:  
Create your own workflow or edit others.
- Workflow management:  
Overview of past and running workflows and jobs.
- Extensible:  
Define jobs with generic Grid Beans or create your own plugins.

# Unicore Rich Client



## Lofar Framework ← Workflow → Unicore

- Two approaches:
  - Remodel everything in existing Unicore content (impractical).
  - Use Lofar Pipeline Framework configuration to create Unicore components.

## Lofar Framework ← Workflow → Unicore

- Reuse components from the Lofar Pipeline Framework.
  - Use config files to create Grid Beans (Job definitions) for Unicore.
  - Extract default argument sets from tasks/generic pipeline parsets.
  - Convert pipeline parsets to unicore workflows.
  - Run Lofar framework subparts in Unicore. Keeping mapfile structure and run jobgroups on single Unicore nodes (questionable).

## ToDo

- Grid Bean creation:
  - Coding a generic Lofar Grid Bean (or two, because of python plugins).
  - Extra parset (for now) with default argument values for each task.
  - Generic Lofar Grid Bean compiled multiple times with different arguments from Lofars “tasks.cfg” (maven project calls from a python script).
  - Create XMLs so server and clients can negotiate possible jobs (“gridbean.xml” for client, automatically during compile; “idb.xml” for server, separate task?).

## ToDo

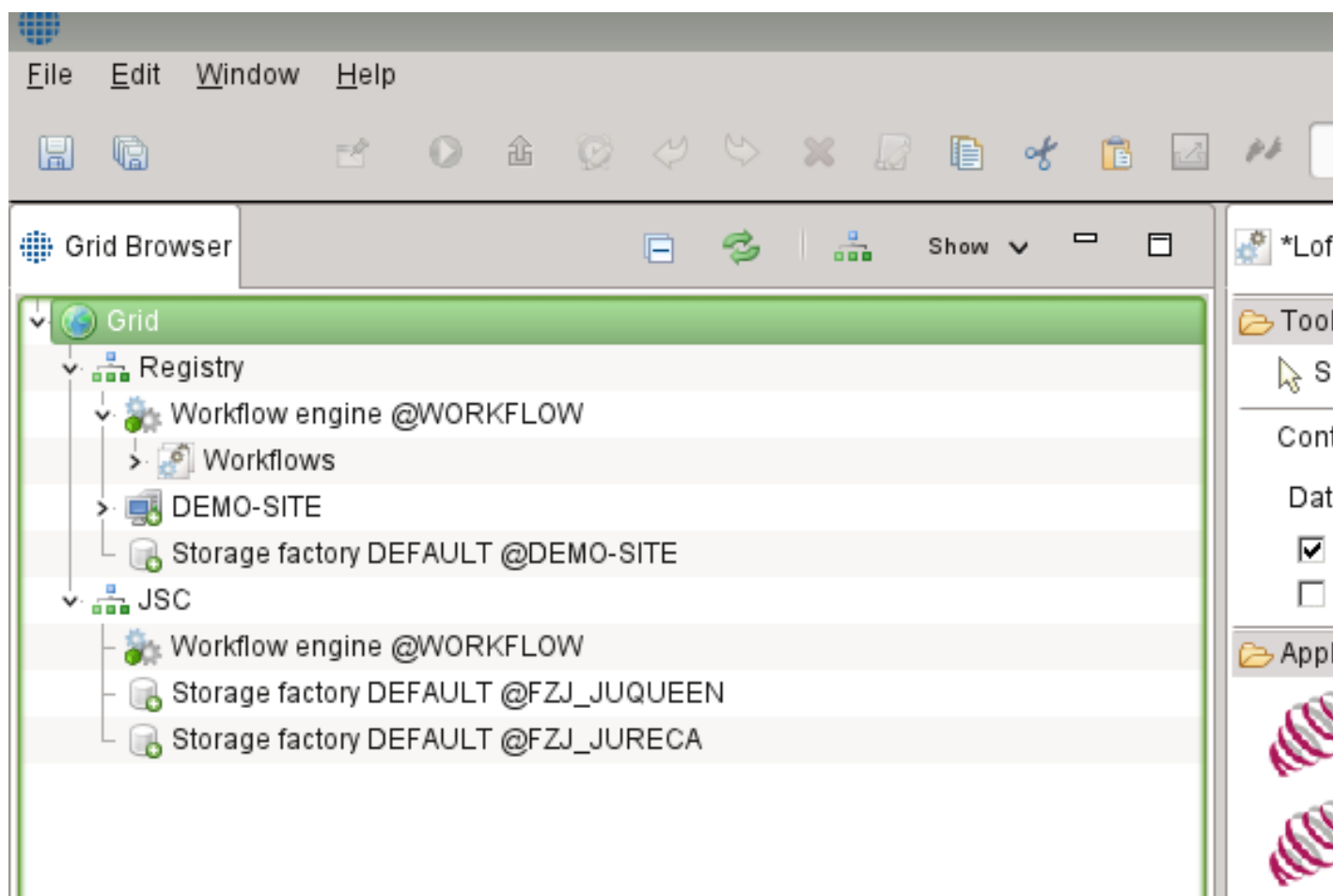
- Server/Management side setups:
  - Provide Lofar software stack
  - Setup Unicore services
  - Distribute standard set of Lofar Grid Beans



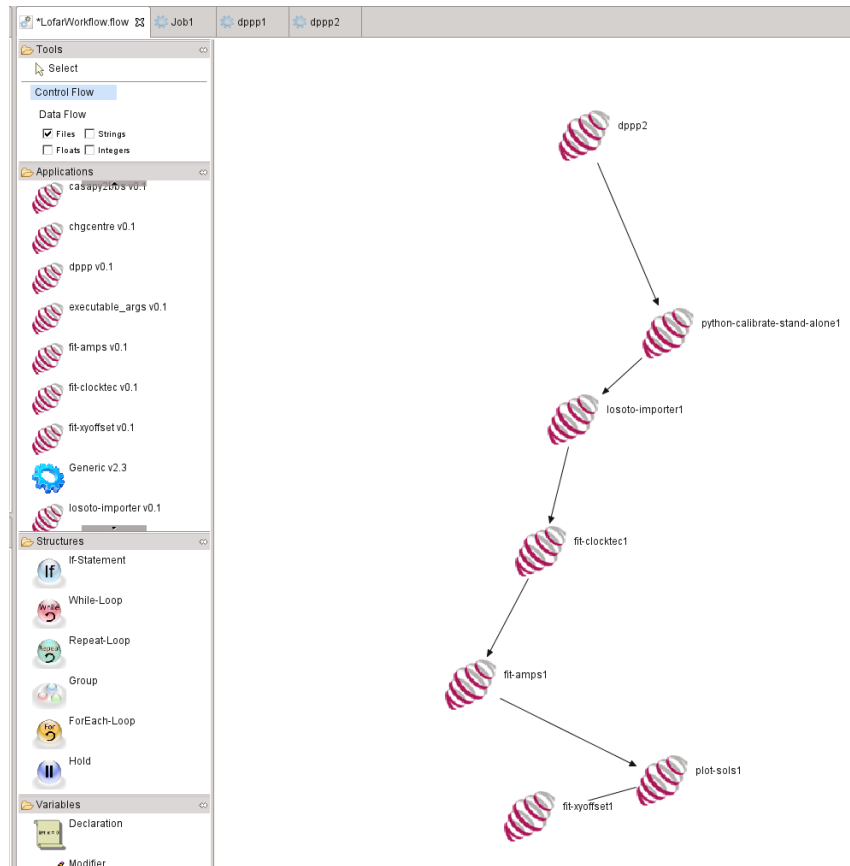
## Status

- Server setup quite “easy” on localhost.
- Job XML creation for server not automatic yet.
- Basic generic Lofar Grid Bean in testing. Format of input definitions near final.
- Started taking care of Data/File management.
- Not looked at automatic pipeline conversions. Needs separate script to convert generic pipeline parsets to Unicore Workflow XML (should be possible).
- Prefactor Pipeline in Unicore ready for testing in a few weeks.

# Grid Browser

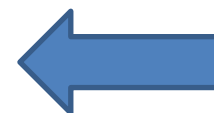
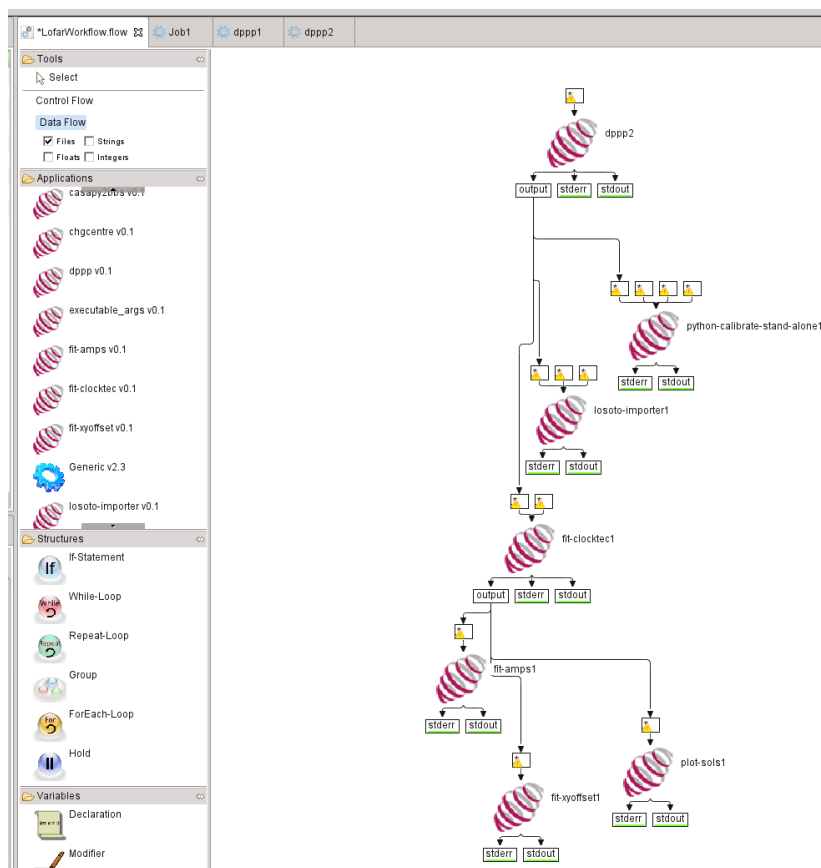


# Workflow Editor Control Flow



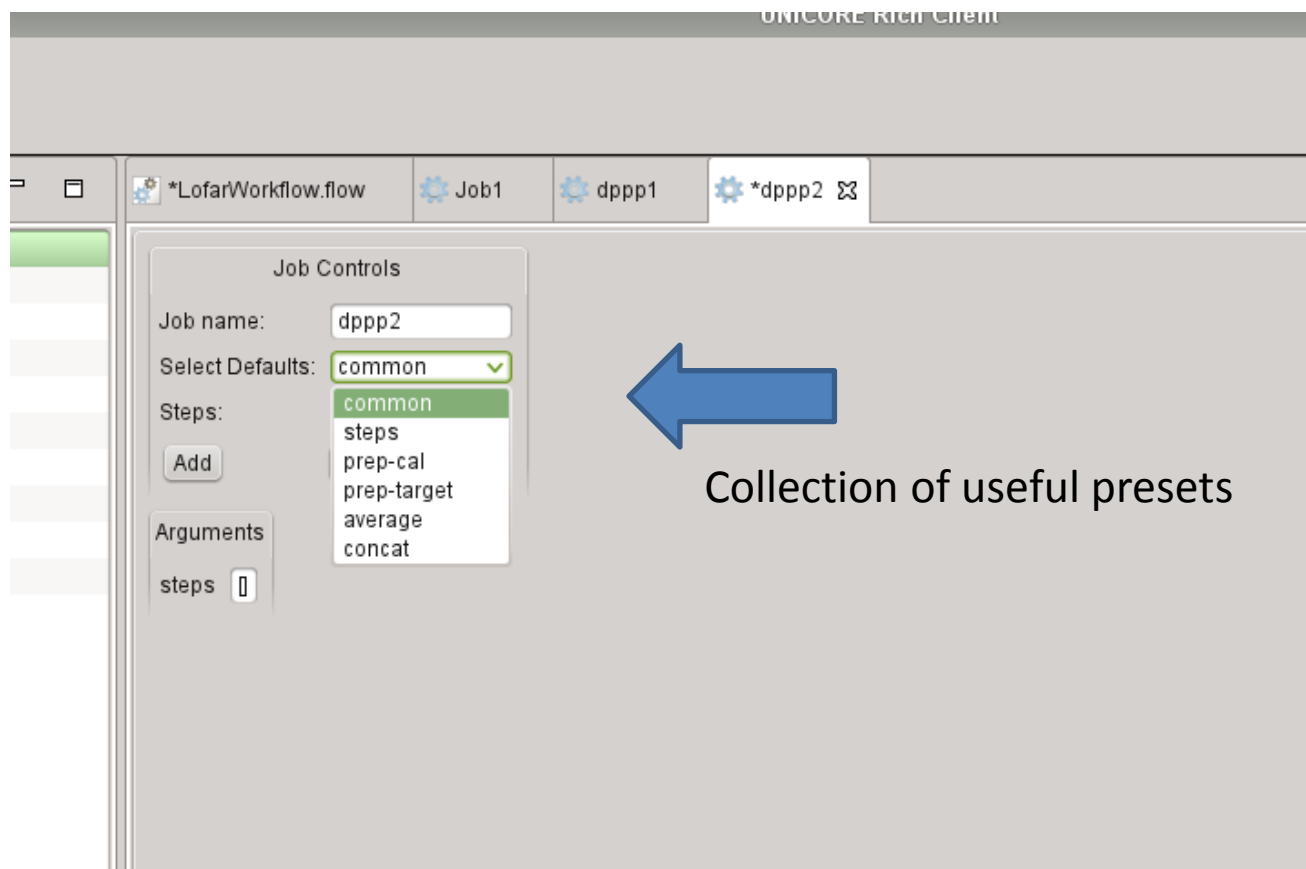
Order of execution.

# Workflow Editor Data Flow

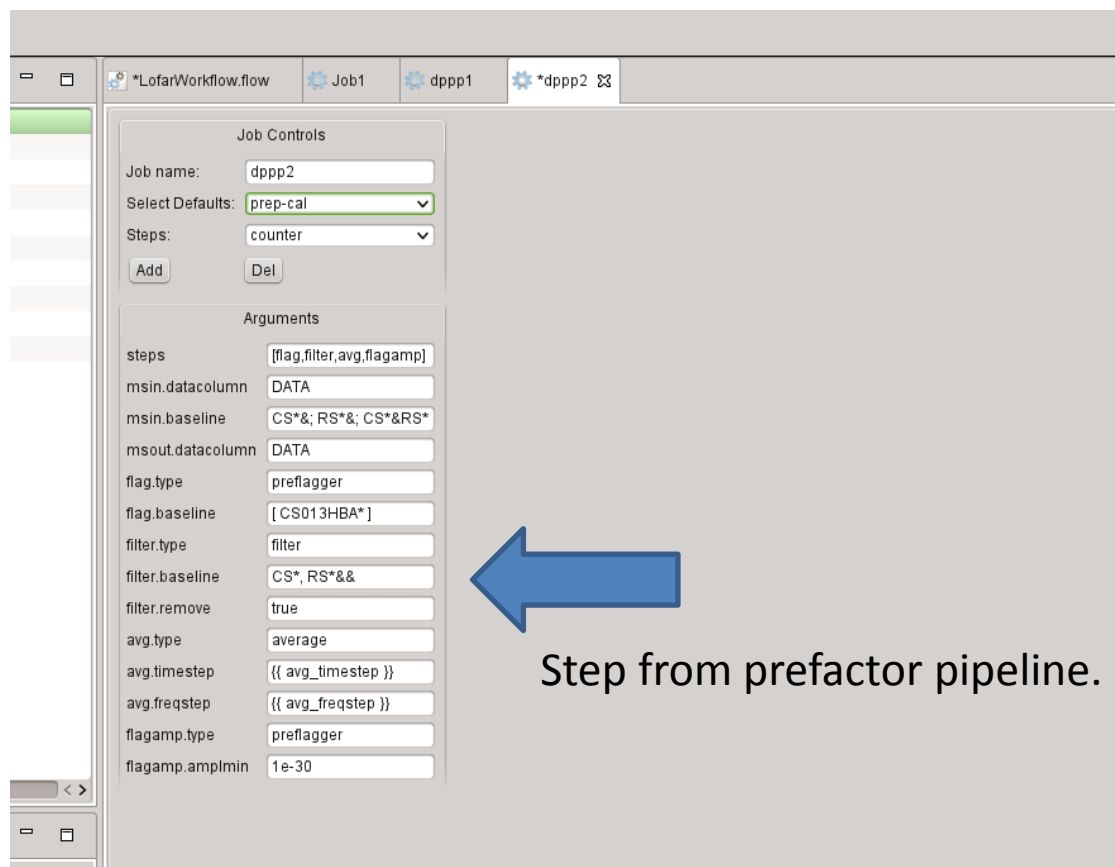


View of inputs and outputs.

## Workflow Editor dppp defaults



# Workflow Editor dppp defaults prep\_cal



The screenshot shows the Workflow Editor dppp interface. The top bar displays the workflow name '\*LofarWorkflow.flow' and the job name 'Job1'. The main panel is divided into two sections: 'Job Controls' and 'Arguments'.

**Job Controls:**

- Job name: dppp2
- Select Defaults: prep-cal (highlighted with a green border)
- Steps: counter
- Buttons: Add, Del

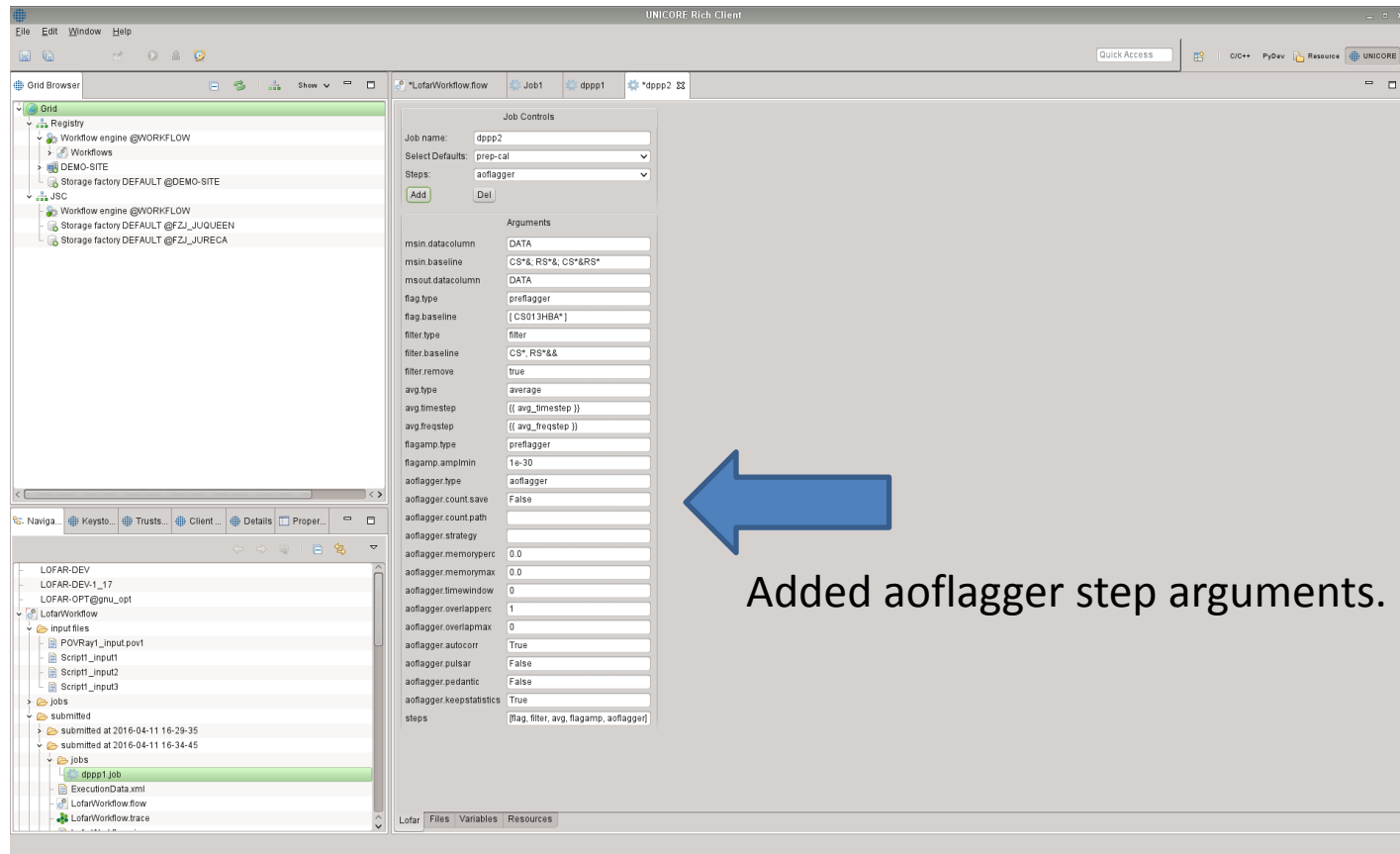
**Arguments:**

steps	[flag,filter,avg,flagamp]
msin.datacolumn	DATA
msin.baseline	CS*&; RS*&; CS*&RS*
msout.datacolumn	DATA
flag.type	preflagger
flag.baseline	[ CS013HBA* ]
filter.type	filter
filter.baseline	CS*, RS*&&
filter.remove	true
avg.type	average
avg.timestep	{{ avg_timestep }}
avg.freqstep	{{ avg_freqstep }}
flagamp.type	preflagger
flagamp.amplmin	1e-30

A large blue arrow points from the text 'Step from prefactor pipeline.' to the 'filter.type' field in the Arguments section.

Step from prefactor pipeline.

# Workflow Editor dppp steps



Added aoflagger step arguments.

# Workflow Editor dppp resource tab

UNICORE Rich Client

Grid Browser

Job Properties:

Use	Property	Value	Unit	Description
<input type="checkbox"/>	Total number of CPUs	1		Total number of CPUs, distributed over all nodes
<input type="checkbox"/>	Number of nodes	1		Total number of computing nodes
<input type="checkbox"/>	CPUs per node	1		Number of CPUs per node
<input type="checkbox"/>	CPU speed	1,024	MHz	Minimal clock rate of the processor
<input type="checkbox"/>	RAM per node	1,024	Mbytes	Minimal amount of physical memory on each computing node
<input type="checkbox"/>	Wall time	60	minutes	Required time on each computing node
<input type="checkbox"/>	OS			Installed operating system
<input type="checkbox"/>	CPU Architecture	sparc		Required processor architecture
<input type="checkbox"/>	Remote login			Login name for the selected resource
<input type="checkbox"/>	Notification email			Email address for notification of successful/failed jobs
<input type="checkbox"/>	Not before			Scheduled start time
<input type="checkbox"/>	Reservation ID			Reservation ID, must be provided by the target site

Selected Target System:

Grid  
Registry  
DEMO-SITE

Resources and machines that provide these.



## Advantages

Unicore could provide a couple of advantages over the current system.

- Ease of use. Intuitive for the user to understand and configure jobs (tooltips, default argument sets, etc.).
- Managing different machines from a central place (Unicore Rich Client). No need to learn different batch system. Possible web frontend in the future.
- Framework development is outsourced. Grid Bean creation comparatively easy (once automatic compiling works).